Engineering Applications of Artificial Intelligence 36 (2014) 303–319

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Ant-based vehicle congestion avoidance system using vehicular networks

Mohammad Reza Jabbarpour [a,*], Ali Jalooli [a], Erfan Shaghaghi [a], Rafidah Md Noor [a], Leon Rothkrantz [b], Rashid Hafeez Khokhar [c], Nor Badrul Anuar [a]

[a] Faculty of Computer Science & Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia
[b] Faculty of Intelligent Systems, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands
[c] School of Computing & Mathematics, Charles Sturt University, WaggaWagga, NSW 2678, Australia

ARTICLE INFO

Article history:
Received 7 February 2014
Received in revised form
11 July 2014
Accepted 3 August 2014
Available online 15 September 2014

Keywords:
Vehicle traffic routing
Ant colony optimization
Vehicular networks
Vehicle congestion problem
Car navigation system

ABSTRACT

Vehicle traffic congestion leads to air pollution, driver frustration, and costs billions of dollars annually in fuel consumption. Finding a proper solution to vehicle congestion is a considerable challenge due to the dynamic and unpredictable nature of the network topology of vehicular environments, especially in urban areas. Instead of using static algorithms, e.g. Dijkstra and A*, we present a bio-inspired algorithm, food search behavior of ants, which is a promising way of solving traffic congestion in vehicular networks. We have called this the ant-based vehicle congestion avoidance system (AVCAS). AVCAS combines the average travel speed prediction of traffic on roads with map segmentation to reduce congestion as much as possible by finding the least congested shortest paths in order to avoid congestion instead of recovering from it. AVCAS collects real-time traffic data from vehicles and road side units to predict the average travel speed of roads traffic. It utilizes this information to perform an ant-based algorithm on a segmented map resulting in avoidance of congestion. Simulation results conducted on various vehicle densities show that the proposed system outperforms the existing systems in terms of average travel time, which decreased by an average of 11.5%, and average travel speed which increased by an average of 13%. In addition, AVCAS handles accident conditions in a more efficient way and decreases congestion by using alternative paths.

1. Introduction

Over the last decade, vehicle population has dramatically increased all over the world. This large number of vehicles leads to heavy traffic congestion, air pollution, high fuel consumption and consequent economic issues (Narzt et al., 2010). In 2010, the American people faced a lot of difficulties due to vehicle congestion which forced their government to spend 101 billion dollars on the purchase of extra fuel (Schrank et al., 2012). Based on a report by Texas A&M Transportation Institute (Schrank et al., 2012), it is estimated that fuel consumption will rise up to 2.5 billion gallons (from 1.9 billion gallons in 2010) with a cost of 131 billion dollars in 2015. Accordingly, finding effective solutions with reasonable cost for congestion mitigation is one of the major concerns of researchers and industries in recent years. Building new, high-capacity streets and highways can mitigate some of the aforementioned problems. Nevertheless, this solution is very costly, time consuming and in most cases, impossible because of space limitations. On the other hand, optimal usage of the existing roads and streets capacity can lessen the congestion problem in large cities at a lower cost.

Intelligent Transportation System (ITS) (Dimitrakopoulos and Demestichas, 2010) is a newly emerged system which collects real-time data for congestion monitoring using road side units (e.g. video cameras, radio-frequency identification (RFID) readers and induction loops) and vehicles as mobile sensors (i.e. in-vehicle technologies or smart phones). These data are used by car navigation systems (CNSs) to find the shortest path or optimal path from a source to a destination. Previous researches (Noto and Sato, 2000; Yue and Shao, 2007; Nazari et al., 2008) concentrated on using static algorithms (e.g. Dijkstra, 1959) and A* (Hart et al., 1968) to find the shortest path in CNSs. Conversely, current studies primarily focus on finding the optimal paths, considering various criteria by utilizing dynamic and meta-heuristic algorithms (Liu et al., 2007; Salehinejad and Talebi, 2008; Boryczka and Bura, 2013). This trend happens due to the dynamic nature of vehicular environments which depends on both predictable and unpredictable events and also because of the

* Corresponding author. Tel.: +60 176238395.
E-mail addresses: reza.jabbarpour@siswa.um.edu.my (M.R. Jabbarpour), ashkansp2@gmail.com (A. Jalooli), erfan_shaghaghi@siswa.um.edu.my (E. Shaghaghi), fidah@um.edu.my (R.M. Noor), l.j.m.rothkrantz@tudelft.nl (L. Rothkrantz), rkhokhar@csu.edu.au (R.H. Khokhar), badrul@um.edu.my (N.B. Anuar).

http://dx.doi.org/10.1016/j.engappai.2014.08.001
0952-1976/

multi-criteria nature of CNSs. Multi-criteria means that distance is not the only objective of CNSs and the drivers. Other key factors include congestion, number of traffic lights, number of route lines, accident risk and travel time. Hansen proved that the multi-criteria shortest path problem is an NP-problem since it requires enumerating all the possible routes (Hansen, 1980).

Recently, Google and Microsoft have predicted vehicle congestion and its duration by performing an advanced statistical predictive analysis of traffic information (Pan and Khan, 2012). This traffic information was provided by the existing infrastructures (e.g. Road Side Units) in order to propose a traffic-aware shortest path for users and drivers. Therefore, this information is not only based on the current traffic information but is also based on other metrics such as weather and historical traffic information. It is worth noting that their system is reactive and avoids vehicle congestion implicitly, which is still not enough, due to the non-recurring congestion. These types of congestion include more than 50% of all vehicle congestion (Coifman and Mallika, 2007). Moreover, the same path is suggested to users by this system and similar to static algorithms, congestion will be switched from one route to another if a significant number of drivers utilize this system. Swarm intelligence algorithms are proposed to solve the aforementioned drawbacks.

Swarm intelligence algorithms are newly emerged algorithms which simulate the behavior of different animals in nature such as ants, bees, fish and birds (Merkle and Blum, 2008; Ahmed and Glasgow, 2012; Yang et al., 2013). These algorithms are able to produce fast, multi-criteria, low cost and robust solutions for various problems such as routing, scheduling and assignment (Merkle and Blum, 2008; Panigrahi et al., 2011). Among these heuristic algorithms, the use of ant-based algorithms has been reported as promising and one of the best approaches for congestion control and traffic management in many research projects (Tatomir and Rothkrantz, 2004; Liu et al., 2007; Dhillon and Van Mieghem, 2007). This paper moves one step forward by presenting a multi-objective ant-based vehicle congestion avoidance system (AVCAS) for proactive congestion avoidance based on real-time traffic information. Our main goal in AVCAS includes proposing the shortest path with the least congestion and travel time, as well as higher vehicle speed. Moreover, non-recurring congestion (e.g. accident, working zones, weather conditions) are also implicitly considered and handled in AVCAS. AVCAS periodically computes n shortest paths, where n is the number of alternative paths, based on the average travel speed prediction and vehicle density for various Origin–Destination (OD) pairs instead of computing these paths for each vehicle (i.e. the number of vehicles is much bigger than the number of OD pairs) and re-routes the vehicles through the least congested path based on their destinations. Implicitly, air pollution and fuel consumption are decreased by this system.

The remainder of this paper is organized as follows: an overview of the ant-based algorithm and its various types and applications are presented in Section 2. Section 3 discusses the ant-based algorithm usage in vehicle congestion control systems. AVCAS and its operation are presented in Section 4, while Section 5 includes the simulation results and system evaluation. Finally, Section 6 concludes the paper and suggests the direction for future research.

2. Ant-based algorithms: definition, types, and applications

How do real ants communicate with each other in finding food sources and accumulate food in their nest using the shortest path, considering the fact that they are blind insects? This question has attracted the attention of many researchers and scientists for many years. The answer is that the ants release a chemical liquid, called pheromone, on their traversed paths based on the quality of the food resource found while moving from their nest to the food source and vice versa. This pheromone trail helps other ants to find the food resources by sniffing the pheromone. The pheromone intensity decreases (pheromone evaporation) over time in order to increase the probability of finding new paths. This phenomenon forms the infrastructure of an Ant System (AS) and Ant Colony Optimization (ACO) algorithms proposed by Dorigo et al. (1991, 1999) and Dorigo (1992) to simulate real ant behaviors by using artificial ants.

Most of the characteristics of real ants are mimicked by artificial ants in order to simulate the behavior of an ant colony for the solution of optimization and distributed control problems. The most common characteristics of real and artificial ants are discussed as follows:

- *Pheromone trail based stigmergy communication*: Pheromone trails assist ants to find the shortest path from nest to food source. While stigmergy communication is a self-organizing behavior of ants which is required to interact with each ant (Theraulaz and Bonabeau, 1999). This communication occurs in an indirect manner which means that an ant alters its surrounding environment by laying pheromone on its traversed paths and the other ants respond to this modification at a later time (Bonebeau et al., 1999). Stigmergy can be transferred to artificial ants by assigning numerical information to the problem space variables and by giving the artificial ants local access to these variables (Dorigo et al., 1999).
- *Implicit shortest path finding*: An implicit shortest path finding happens by reinforcement on the shortest path for both real and artificial ants. More pheromone is laid on the shortest paths because they are completed more faster than longer paths (Di Caro, 2004).
- *Concurrent and independent iterations*: The artificial individual ant, similar to the real one, is able to find a path from nest to food, but it is not the only ant that does this task. The other ants do the same task concurrently and independently in order to converge to the optimal path in a short time (Dorigo and Birattari, 2010).
- *Discrete world*: Artificial ants, unlike real ants, live in a discrete world which means that their actions are transitions from one discrete condition to another (Dorigo et al., 1999).
- *Synchronized vs. desynchronized system*: Artificial ants move from their nest to the food source and vice versa in each iteration. Therefore, they move in a synchronized way unlike real ants which move in a desynchronized pattern (Blum, 2005).
- *Memory*: The artificial ant utilizes an embedded memory to store the traversed path information. It is used for building and evaluating possible solutions, for backtracking from destination to source and for updating the pheromone value on the found path. In comparison, real ants do not have memory but use their sensing capability for this purpose.
- *Pheromone evaporation strategy*: Pheromone evaporation happens very slowly in nature and its rate is constant (Deneubourg et al., 1990). This mechanism and its evaporation rate vary from one problem to another in the simulation environment for artificial ants.
- *Extra capabilities*: Artificial ants use extra capabilities to increase the efficiency of the whole system that can be augmented with capabilities such as future prediction, local optimization, and backtracking, while these capabilities cannot be found in real ants.

2.1. Ant colony optimization (ACO)

In this section, we describe the procedure of ACO algorithm. Even though many changes have been applied to the ACO

algorithm in the last two decades, its basic mechanism and steps are still remained unchanged. The steps of the ACO algorithm are as follows:

1. *Problem graph depiction*: Artificial ants move from one discrete state to another. Therefore, they can solve discrete problems (LaValle, 2006) which can be depicted as a graph with N nodes and L links.
2. *Initialization*: A number of artificial ants (N_a) are located on each node (source) and a specific value (weight) is assigned to each link of the problem graph. The re-generation period of ants (γ), which means that the Time Interval (TI) between two consecutive ant colonies generation is started periodically at a pre-defined TI for finding new paths. N_a and γ are obtained using experiments or trial and error. Moreover, physical distance, random number, queue length or a number obtained by mathematical formula can be used as initial weight of links. The node transition rule is defined and used for next node selection. The probability of choosing j as a next node from i by ant k is calculated by (Dorigo et al., 1991; Dorigo, 1992)

$$P_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{h \notin tabu_k} (\tau_{ih})^\alpha (\eta_{ih})^\beta} & \text{if } j \notin tabu_k, \\ 0 & \text{otherwise}, \end{cases} \quad (1)$$

   Intensities α and β are the relative importance that can be used to stress the importance of pheromone intensity τ_{ij} and η_{ij} route cost. $tabu_k$ is the set of visited nodes by ant k.
3. *Pheromone update*: The ants start to move from source to destination using the node transition rule (Eq. (1) and store visited nodes in their memory. Whenever an ant reaches its destination, it backtracks to its origin/source node using its memory and updates the links pheromone value on its return path using the pheromone update rule. Two concepts are embedded and considered in this rule. On one hand, the pheromone value of the links which are not traversed by the ant should be decreased in order to reduce the probability of their selection by the other ants. This issue is called pheromone evaporation and should be considered in the pheromone update rule. On the other hand, the pheromone value of the links which are traversed by the ant should be increased in order to increase the probability of their selection by the other ants and this is called pheromone reinforcement in the ACO algorithm.

   If the pheromone value decreases slowly, the ants will be trapped in suboptimal solutions in most of the cases. However if it decreases quickly, the ants will not take advantage of data gathered by the other ants. Therefore, the pheromone evaporation rate has a direct impact on the exploration and exploitation of paths in the ant-based algorithms. Exploration means finding a new path, whereas exploitation means improving the current found path. Assigning an appropriate pheromone evaporation rate is needed in order to set a proper trade-off between these two factors. More discussion on this issue can be found in Claes and Holvoet (2011). The pheromone update rule which includes both pheromone evaporation and reinforcement phases is given as (Dorigo et al., 1991; Dorigo, 1992)

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^k, \quad (2)$$

   where $\rho \in (0, 1]$ is a constant value, named pheromone evaporation, and m is the number of ants. The amount of pheromone laid on links i and j by ant k is calculated using

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{f_k} & \text{if the } k\text{th ant traversed link } (i,j), \\ 0 & \text{otherwise}, \end{cases} \quad (3)$$

   where Q is the constant value and f_k is the cost of found route by ant k.
4. *Stopping procedure*: The ACO algorithm is completed by reaching a predefined number of iterations whereas an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination.

2.2. Types of ACO and its applications

The first study to adopt the ant system (AS) and the ACO algorithm aims to solve the salesman problem by introducing the Traveling Salesmen Problem (Dorigo et al., 1991; Dorigo, 1992). Promising results in its initial experiments and also its novelty attracted researchers attention and a number of extensions of the principle were introduced in recent years that could improve the ASs performance in a significant way. These extensions include elitist AS (Dorigo, 1992), rank-based AS (Bullnheimer et al., 1997), and MAXMIN AS (Stützle and Hoos, 2000). The main differences between the initial AS and these extensions are the pheromone update procedure and some additional details in the pheromone trails management. There are some other ACO algorithms that modify the features of the AS in the literature. More details about these approaches and their differences can be found in Dorigo and Birattari (2010) and Dorigo and Stützle (2010).

In addition to proposing different extensions of the ACO algorithm, ACO has been widely utilized in different research areas and industries in recent years. Computer, electronic, civil and mechanical engineering are the most predominant domains that receive more benefits from ACO. Routing algorithms (Di Caro and Dorigo, 2011; Misra et al., 2010), digital image processing (Siang Tan and Mat Isa, 2011; Tian et al., 2011), bridge piers design (Martínez et al., 2010, 2011) and scheduling (Chen et al., 2010; Berrichi et al., 2010) are some areas which have used ACO to improve efficiency. More information about ACO applications can be found in Dorigo and Stützle (2010) and Chandra Mohan and Baskaran (2012).

ITS merges computer network-based information (e.g. vehicular networks and wireless sensor network) and electronic technologies (e.g. sensors and cameras) with transportation technologies to provide vehicle routing and congestion control mechanisms. Recently, studies have augmented the ACO algorithm into vehicle routing and congestion control systems obtaining promising results. These systems are discussed in the following section.

3. Progress of ant-based vehicle congestion control systems

Before describing an ant-based vehicle congestion control system, it is worth noting that most of the recent approaches and solutions in vehicle traffic domain utilize the collected traffic data via Global Positioning Systems (GPS), Geographical Information Systems (GIS), Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications to predict the future traffic information on the roads (Gong et al., 2008). Although prediction of traffic information such as vehicle density, travel speed and time are important factors for vehicle traffic routing, it is a difficult process due to various unpredictable events and dynamic nature of vehicular environments. Neural networks are utilized by some researchers (Shen, 2008; Van Lint et al., 2005; Dia, 2001; Park et al., 2011) for vehicle travel time and speed prediction due to their learning capabilities. In these approaches, proposed algorithms trained with historical traffic data and used real-time traffic data for vehicle travel time and speed prediction. Vehicle travel time prediction is used in most of the existing approaches. Predicted data can be used for appropriate vehicle routing and congestion avoidance.

Traffic congestion recognition and avoidance approach by using vehicular networks is proposed by Wedel et al. (2009). Average travel speed of vehicles is used for congested road detection and congestion avoidance. Congestion avoidance and route allocation using virtual agent negotiation (CARAVAN) (Desai et al., 2013) is a multi-agent system that is developed for finding optimal paths within a short time and with low communication overheads. Vehicles exchange preference information and use virtual negotiation for collaborative route allocation through inter-vehicular communications in CARAVAN. Combination of the Dijkstra algorithm and a heuristic algorithm (e.g. Particle Swarm Optimization, Simulated Annealing or Tabu Search) is proposed as a hybrid vehicular re-routing strategy for vehicle traffic congestion avoidance by Wang et al. (2013). Three re-routing strategies, namely multi-path load balancing consider future vehicle positions (EBkSP), random multi-path load balancing (RkSP), and Dynamic Shortest Path (DSP), are presented by Pan and Khan (2012), for vehicle congestion avoidance that use V2I communications for real-time data collection. Batool and Khan (2005) proposed another travel time and traffic congestion prediction method using ad hoc networks. They developed a multilayer feed forward neural network combined with a back-propagation algorithm (Rumelhart et al., 1988) for traffic congestion prediction. Since our proposed approach, AVCAS, is based on an ant-based algorithm, we present a comprehensive overview of ant-based vehicle congestion control systems in the following paragraphs.

Initially, Fan et al. (2004) proposed an ant-based algorithm to solve the shortest path problem without considering vehicle congestion. Since the ant-based algorithm was proposed and examined for static problems (e.g. shortest path problem), researchers have started to improve the steps or found optimal values for various parameters of the ACO to prepare it for dynamic problems such as vehicular environments. Some limited studies have been conducted in finding the best values for the parameters in ACO to reduce vehicle congestion on the roads (Liu et al., 2007; Nahar and Hashim, 2011; Ok et al., 2011). On the other hand, most of the efforts were performed on modifying the ants performance or the probability function in the second step of ACO. For instance, a new type of ant, namely the check ant, was introduced to preserve better routes and discard degraded ones by considering the travel time as an evaluation metric (Ghazy et al., 2012). City agent, road supervisor agent and intelligent vehicle-ant agent are three different ants which were used to find the best route, based on real-time data from a vehicular network (Kammoun et al., 2010). In addition, other studies (Ge et al., 2011; Bedi et al., 2007; Hallam et al., 2004) described modified probability functions for the next hop finding. However, most of the probability functions are similar to the basic probability function; the only differences are minor modifications such as the addition or omission of parameters from the basic probability function.

In the second stage of the research, road map segmentation was utilized by Narzt et al. (2010), Tatomir and Rothkrantz (2006) and Claes and Holvoet (2012) to overcome dynamic and quick changes of vehicular environments. A hierarchical routing system (HRS) based on the ant algorithm that splits a road map into several smaller and less complex maps was proposed (Tatomir and Rothkrantz, 2006). HRS considers only historical and present traffic conditions and does not pay attention to future conditions (Tatomir et al., 2009). The ant-based algorithm was used to guide a vehicle at each intersection by creating routing tables. Narzt et al. (2010) introduced another technique that uses segmentation as a principle to overcome traffic control problems. In this approach, a novel pheromone update with a user preference assignment system was adopted to divide the environment into different clusters. This approach adopts only the pheromone mechanisms without considering real-time information that can be collected by new technologies (Kanamori et al., 2012). A cooperative ant-based algorithm was proposed, based on the region concept (Claes and Holvoet, 2012) where near segments are grouped together to form a region. Thereafter, routing is performed according to the regions instead of segments. However, maintaining the additional data needed for each specific region presents a high overhead to this approach (Dias et al., 2013).

Besides segmentation, prediction based congestion control systems were proposed which utilized the traffic patterns or traffic condition prediction to reduce congestion on the road. Link travel time or vehicle queue length prediction based on real-time information was used for congestion prediction and reduction (Claes and Holvoet, 2011; Tatomir et al., 2009; Ando et al., 2006). Artificial intelligence algorithms were adopted, such as machine learning, neural networks and fuzzy logic to predict the traffic pattern. Yousefi and Zamani (2013) proposed an optimal routing method using a machine learning algorithm to reduce vehicle travel time by combining ACO and learning approaches based on the length of paths. Jiang et al. (2007) proposed the shortest path finding method by modifying the pheromone update rule and adding a learning strategy into the ACO. In the case of neural network and fuzzy logic, dynamic route selection was proposed by Abbas et al. (2011) and Salehinejad and Talebi (2010) to avoid traffic congestion. Fuzzy logic and neural network were utilized for pheromone update and traffic prediction, respectively. Although using traffic prediction can improve the congestion avoidance procedure, it explicitly leads to the identification of sub-optimal paths instead of optimal paths in most cases (Dooms, 2013). Moreover, using artificial intelligence techniques for traffic prediction may increase both system overhead and complexity which is not suitable for vehicle traffic routing systems. A comprehensive review of ant-based vehicle traffic routing systems can be found in our previous study (Jabbarpour et al., 2014).

Although most of the reviewed approaches obtained promising results for vehicle congestion control, we believe that combining segmentation with short-term prediction and avoiding congestion instead of recovering from it is a better solution for congestion problems in a vehicular environment. In addition, although accidents are one of the main reasons for vehicular congestion (Wang, 2010), they were not considered in most of the current approaches due to their complexity and unpredictability. However, AVCAS which implicitly considers accidents in its routing process is proposed in this paper to achieve congestion avoidance. This algorithm uses segmentation, the average travel speed of links as a short-term prediction metric, vehicle density and average travel time to accomplish this goal.

4. Ant-based vehicle congestion avoidance system (AVCAS)

Before explaining the various steps of AVCAS, we discuss the technical architecture that AVCAS uses and whether it is centralized or decentralized. In a decentralized architecture, route finding and computation takes place for each vehicle individually using the on-board processor and memory of that vehicle. It is ideal if vehicles receive traffic information through wireless communication (e.g. vehicle-to-vehicle or vehicle-to-infrastructure) and include road maps and GPS. In a centralized architecture, route finding and computation takes place through a central server in response to requests from drivers. In this architecture, the central server has access to the historical or real-time traffic information database and computes routing algorithms based on this information. More information about centralized or decentralized architectures and their pros and cons can be found in Suson (2010).

AVCAS is designed for decentralized architecture, however the architecture is not yet available to be used, since all vehicles are

not equipped with on-board units and wireless transceivers. AVCAS can be configured to be distributed over several on-board vehicle navigation systems, but at this moment, it can only be executed in a centralized manner. Therefore, AVCAS is part of a group of distributed centralized servers (see Section 4.1 for more details) which provides route advice to the drivers who are equipped with transceiver devices (e.g. mobile phones, PDAs, Tablets). Our model also assumes that the system can detect the position of the vehicles via GPS. AVCAS is expected to guide vehicles through the least congested shortest paths to their destinations. As was discussed previously, problem environment depiction, initialization, pheromone update and stopping procedure are the main steps of ant-based algorithms (see Section 2.2). Each of these steps in AVCAS is discussed in the following subsections.

4.1. Problem environment depiction

In order to find border nodes and overall overview of road map, and to reduce system overhead and complexity, problem environment depiction includes two phases, namely segmentation and layering, which are explained in the following. Using a distributed centralized management system (instead of one centralized system) is another advantage of these phases. Our proposed layered and segmented model which consists of the following four different bottom-up layers is illustrated in Fig. 1 and explained as follows:

1. *Physical layer*: This layer shows the real road map and nodes corresponding to intersections and junctions with links corresponding to streets and highways. This map can be exported from map databases like OpenStreetMap. In this layer, the road map is converted to a graph and this graph is given by $G_p=(N_p, L_p)$, where N_p and L_p are the set of nodes and links, respectively.
The segmentation phase happens in this layer and divides road map into number of segments with different sizes. Specifying the segments size is based on the number of nodes (i.e. junctions, intersections) in each segment. In other words, their sizes are assigned in such a way that there is approximately identical number of nodes in each segment. Different sizes of segments are considered in order to maximize the use of resources such as processors and storage devices, and also to balance and reduce the routing overhead in different segments. Moreover, dynamic and quick changes of vehicular environments can be managed using map segmentation and routing is accomplished for each segment individually instead of the whole map. This segmentation is applied in physical layer and each segment is managed by one navigator. Navigators are responsible for creating and updating the routing table for their own segment, which is called the Intra Segment Table ($Intra{-}ST_{(i)}$), using an assigned weight to each link in the graph based on the pheromone update rule (Eq. (11)), where i is the segment number (or identifier). Besides, Table of Segment (*ToS*) is created in segmentation phase which includes the nodes' name, ID and their segment in order to give a complete road map view to all navigator servers. This table is distributed among all servers and helps them to detect the destination nodes' segments in the routing procedure.
One of the main requirements of CNSs, using decentralized processing system, is achievable through the segmentation phase. It means that several centralized servers (navigators) are distributed throughout the segmented road map (i.e. one server for each segment), namely distributed centralized servers, to guide the vehicles to their destinations. In this way, the road map searching time and the size of routing tables are reduced significantly. This is because each navigator is released from maintaining the whole map information and maintains only small routing tables with local information. Using different strategies (i.e. different types of ants, pheromone update rules, pheromone trail laying rules and probability functions) for different segments is another advantage of road map segmentation in AVCAS.
2. *Junction layer*: Ineffective nodes which do not correspond to a junction are eliminated on the junction layer due to identification of an impossible turning. Junctions are the most critical points in the vehicle routing process.
3. *Border nodes layer*: Junctions and their links which connect two different segments in junction layer are retained, otherwise, they are pruned. The remaining nodes (junctions) are called the border nodes. The Border Nodes Table ($BNT_{(i)}$), which stores border nodes information, is created for each segment and used for routing to surrounding segments. The BNT of each segment is disseminated among all junctions of same segment. Thus, this layers information can be used whenever the source and the destination of a vehicle are not in the same segment, but there is a direct link between them.
4. *Routing layer*: The information in this layer is used whenever a vehicle travels over long distances and thus traverses more than one segment to reach its destination. To achieve this goal, a node is assigned to each segment and a link is added between two nodes if there is a link between these two segments' border nodes. An Inter Segment Table (*Inter-ST*) is created based on this new graph (G_R) and is used for distant destination routing (e.g. between segments). $G_R=(N_R, L_R)$, where N_R is the set of nodes assigned to each segment, therefore their numbers are equal to the number of segments, and L_R is the set of links between N_R. The *Inter-ST* is disseminated among all segments

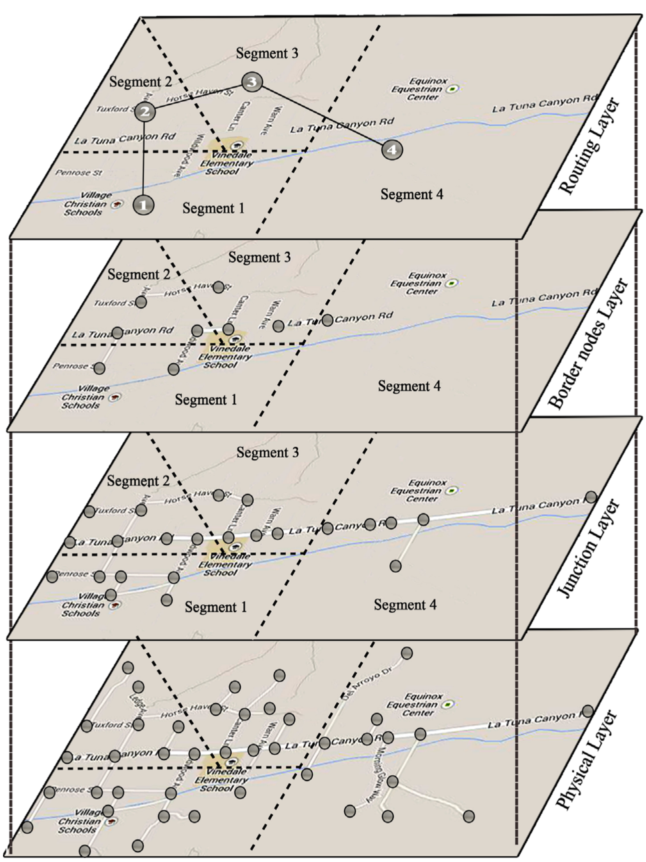

Fig. 1. Proposed layered and segmented model for AVCAS.

navigators to give an overall view of the map to the distributed navigators.

Three different cases may occur in the routing process of a vehicle from a node I (source) in X segment to node y (destination) in Z segment:

1. the source and destination nodes are in the same segment;
2. the destination node is within one of the source nodes surrounding areas;
3. other (neither case 1 nor 2).

It is worth noting that servers detect the segments of sources and destination nodes through *ToS*. Therefore, in case 1, the $Intra-ST_{(x)}$ is used by the navigator for each routing decision. In case 2, the $Intra-ST_{(x)}$ is used to guide the source node to a border node of the same area. Then the $BNT_{(x)}$ is used to guide the source node to a border node of the destination segment. After that, the $Intra-ST_{(z)}$ is used to guide the source node to a destination node within the destination area. In case 3, a similar strategy as in case 2 is used, however the *Inter-ST* is used to guide the source node from the previously selected border node to the proper border node of the next segment until it reaches its destination segment. All of these routing tables (i.e. $Intra-ST_{(i)}$, $BNT_{(i)}$ and *Inter-ST*) are updated using the ant-based algorithm which is discussed in following section.

4.2. Initialization

Ants are the cornerstone of our system in the initialization phase and we discuss the various types of ant agents used in AVCAS in this subsection. Vehicle as ANT (VANT) and Packet as ANT (PANT) are the two main types of ant agents modeled. PANTs are further divided into two types, namely Forward ANT (FANT) and Backward ANT (BANT).

4.2.1. Vehicle as ANT (VANT)

Vehicles are used as real ants in our system in order to collect accurate real-time information using VANET infrastructures. Vehicles send basic data such as *ID*, time and direction as a message to road side units, which are located at junctions. Using these data, the navigation servers predict the travel speed of each link (PTS_{ij}) in its segment based on historical and real-time speed information. Based on Kwon and Petty (2005), Nanthawichit et al. (2003), and Chien and Kuchipudi (2003), both current traffic conditions and historical data are required for accurate short-term prediction of the time it takes for a driver from an origin to arrive at a destination. Because vehicles' traveling time, speed and density or in other words the cost or probability of each road/street on the road map are not constant over time, for this prediction, we add a vertical time axis and break the time into discrete intervals $(I_0, I_1, \ldots, I_n)$ (i.e. 10 s), where $I_k = [start_k, end_k]$ (i.e. discrete TI beginning at time instant $start_k$ and ending at time instant end_k) in order to consider the dynamic aspect of these data. As a result, our proposed links probability function (Eq. (7)) for Forward ANTs (FANTs) in the next section is a discrete and time-dependent function and a set of time-dependent cost or probability is assigned to edges (roads). A discrete time dynamic network can be represented as a static network using a time-expanded network model (Köhler et al., 2002), which is a useful implicit tool for visualizing, formulating and solving discrete time dynamic shortest path problems. By adopting this model, probability or cast can be satisfactory approximated for each interval. Historical Travel Speed (HTS) and Current Travel Speed (CTS) are calculated and assigned to each link (i,j) for each TI. If we assume that $t=n$ is the current time, CTS and HTS for link (i,j) at time n are calculated

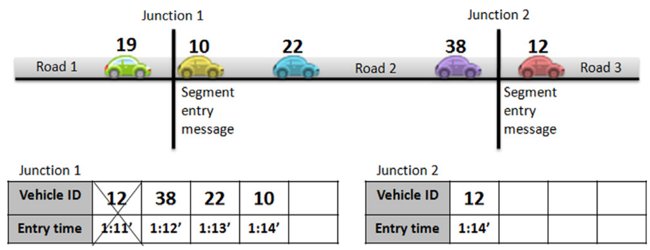

Fig. 2. Procedure for current average travel speed calculation in AVCAS.

using Eqs. (4) and (5), respectively, as follows:

$$CTS_{ij}^{n} = \frac{LL_{ij} \times NV_{ij}}{\sum \Delta t_{ID}}, \tag{4}$$

$$HTS_{ij}^{n} = \frac{\sum_{t=1}^{t=n-1} CTS_{ij}^{t}}{n-1}, \tag{5}$$

where NV_{ij} is the number of vehicles on link (i,j), and LL_{ij} is the length of link (i,j) and has a constant value. Δt_{ID} is the time duration used by specific vehicle (ID) to traverse a road between two consecutive junctions. The navigation server of each segment is able to calculate Δt_{ID} and NV_{ij} for its segments links (roads or streets) as illustrated in Fig. 2.

NV_{ij} is obtained from the number of vehicle IDs in the table of junction i (NV_{ij} is 3 in our depicted example). Vehicles' information is omitted from the junction table upon reaching a new junction and is used by the navigation server to calculate Δt_{ID}. In the example in Fig. 2, the information of the vehicle whose ID of 12 was omitted from junction 1's table and Δt_{12} is calculated as $1:14-1:11$ min $=3$ min. The Short-term Predicted Travel Speed (PTS) of link (i,j) at time $t=n+1$ is computed as follows:

$$PTS_{ij}^{n+1} = \xi(HTS_{ij}^{n}) + \lambda(CTS_{ij}^{n}) \tag{6}$$

where ξ and λ weight the effect of historical and current travel speeds on the predicted travel speed of roads respectively.

It is worth noting that both HTS and CTS are used for PTS calculation in order to consider both recurring (regular) and non-recurring congestion, respectively. Delay on a road can be caused by incidents for example by accidents (non-recurring congestion or delay). But there are also regular delays in the rush hours. Regular (recurring) delays exist in historical data. It is important that our system be sensitive to both of these delays and adapts immediately. That is why Eq. (6) includes current speed and the regular speed from the historical data. In the non-recurring delay cases that the delays differ completely from the historical data, the adaptation is suboptimal by taking care of historical data. But in most cases the delay will be similar to the historical data so it is good that our system anticipates the regular delays from historical data even if the current speed does not show any delay. Taking care of only current speed reduces the adaptability of our system. For example, if one driver drives slower than the normal speed it will affect the other vehicles speed as well since its rear vehicles try to change their lane and this lane changing will also affect the vehicles speed on the other lanes. However, since this speed reduction is instant and temporary its impact on the routing system should be reduced. This can be obtained by considering historical data because real-time data on its own is not adequate for mitigating this issue.

4.2.2. Packet as ANT (PANT)

After the gathering of real-time information by VANTs, PANTs are used to explore and find the shortest least congested path between source and destination. FANTs and BANTs are used for this purpose. We have improved the ant packet header which was proposed in AntNET (Dorigo and Sttzle, 2004) due to the changes

Packet Type	Address of source node	Segment of source node	Address of destination node	Segment of destination node	Packet length	Packet Sequence Number	Packet Start Time	Packet's Memory	Size of Memory

Fig. 3. Improved ant packet header.

in problem environment depiction (i.e. segmentation and layering). Two new fields are added to represent the source and destinations segments. The new header is used by PANTs during road map exploration and path finding procedure. This packet header is illustrated in Fig. 3.

(a) Forward ANT (FANT): FANTs explore the road map and construct routes between two specific points (source and destination). FANTs build a solution by choosing probabilistically the next node to move forward. Eq. (7) represents a new probability function in our system which is used by FANTs:

$$P_{ij} = \frac{\alpha(\tau_{ij}) + \beta(\eta_{ij})}{\sum_{h \in tabu_k}(\alpha(\tau_{ij}) + \beta(\eta_{ij}))} \times \left(\frac{1}{1 + \frac{1}{N_j}} \right), \tag{7}$$

where τ_{ij} (pheromone value) is the learned desirability for an ant in node i to move to node j (next hop) and is computed by BANTs using Eq. (11). η_{ij} reflects the instantaneous state of the vehicle density and velocity on the link from i to j and computed by VANTs. α and β weight the importance of τ_{ij} and η_{ij}, and are called pheromone power and real-time information power in this paper, respectively. In other words, the impact of gathered data by PANTs and VANTs is tuned by α and β. $tabu_k$ is the set of candidate nodes connected to node i that an ant has not visited yet. Finally, N_j represents the number of neighbors for node j. The number of neighbors of next hop, N_j, is considered in Eq. (7) in order to give higher priority to a node with more neighbors. In this way, the probability of finding new paths increased. FANTs do not deposit any pheromone while moving and do not go to other segments. This allows the use of different types of ants for different segments. The calculation methods of τ_{ij} and η_{ij} are explained in the following sections.

η_{ij} is computed using Eq. (8) by considering vehicle density (D_{ij}) and their average predicted travel speed (PTS_{ij}) on link (i,j). These two factors are chosen due to their predominant role in vehicle congestion navigation:

$$\eta_{ij} = (1 - D_{ij}) + \left(\frac{PTS_{ij}}{\varphi} \right), \tag{8}$$

where φ is assigned 80 km/h ($\simeq$ 22 m/s) which is the maximum speed limit for urban and town area in Malaysia (Ong et al., 2011). PTS_{ij} is divided by φ to avoid obtaining large values for Eq. (8). In addition, D_{ij} is calculated using Eq. (9) as follows:

$$D_{ij} = \frac{NV_{ij}}{Max\_NV_{ij}}, \tag{9}$$

where NV_{ij} is the number of vehicles on link (i,j), and Max_NV_{ij} is the maximum capacity of the link (i.e. maximum number of vehicles which can be on the road simultaneously in congested condition) and computed using

$$Max\_NV_{ij} = \frac{LL_{ij}}{L_V + \Delta L} \times NL_{ij} \tag{10}$$

where LL_{ij} and NL_{ij} are the length and the number of lanes of link (i,j), respectively. ΔL is the average space between two consecutive vehicles. Finally, L_V is the average length of vehicles. ΔL and L_V are considered as 2 m and 5 m in this paper, respectively (Cheung et al., 2005).

(b) Backward ANT (BANT): When a FANT reaches its destination, it changes its role and becomes a BANT, instead of dying and copying its memory to BANT (i.e. which happen in most of the ant-based algorithms). In this way, the time complexity of the overall system is reduced. The BANT returns the same path as the one traversed by FANT by using its memory but in the reverse direction. The BANT updates the links' pheromone intensity using the pheromone update rule, which is discussed in more detail in next subsection.

4.3. Pheromone update

When all the FANTs have reached their destinations, the pheromone level of each link is updated. This update can either increase or decrease the pheromone trial values. These two phases are called pheromone reinforcement and evaporation in ant-based algorithms. BANTs use the FANTs memory to return from the destination to the source node. Therefore, they can evaluate the cost of the solutions that they generate and use this evaluation to modulate the amount of pheromone that they deposit on the links in return mode. Making pheromone update a function of the generated solution quality can help in directing future ants more strongly toward better solutions. In fact, by letting ants deposit a higher amount of pheromone on the optimal paths, the ant's path searching is more quickly biased towards the best solutions. The intensity of pheromone is increased or decreased by using Eq. (11), which is called the pheromone update rule in our system:

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k}, \tag{11}$$

where $\rho \in$ (0, 1] is a constant value, named pheromone evaporation, and n is the number of nodes in the desired segment. The amount of pheromone laid on links i and j by ant k is calculated using

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{TT_{ij}^{k}} + \frac{1}{D_{ij}^{k}} + \frac{1}{LL_{ij}^{k}} & \text{if the } k\text{th ant traversed link } (i,j), \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where TT_{ij}^{k}, D_{ij}^{k} and LL_{ij}^{k} are the travel time, vehicle density and length of each link of the found route by ant k, respectively. As a result, if a link belongs to a found route by an ant, its pheromone value is increased (reinforcement) considering its travel time, vehicle density and length. If it does not belong to a found route, its pheromone value is decreased (evaporates). It means that Eq. (11) first decreases the pheromone value of all links and then increases it for the links belonging to the found route. Pheromone evaporation improves the exploration factor of the search and encourages the ants to find new routes instead of insisting on the first found route.

Most of the current approaches do not consider accidents in their system due to the complexity of these unpredictable events. In order to consider accidents, we have to deal with two main points: First, the accident, that is, its accurate position, time and status (i.e. how serious it is) should be detected based on the collected data via sensors and video cameras or received through reports from other drivers via their smart phones or devices. Second, finding a way to update the routing tables in the shortest time possible with the least delay on the accident condition is a necessity. It is worth noting that some of the current approaches

such as the HRS (Tatomir and Rothkrantz, 2006), disable (i.e. ignore) the road where an accident has happened for a while and let the system perform without considering that road in the routing process. The point with the ant-based algorithm is that it handles this situation with some delay due to the stochastic feature of the searching approach and this causes many vehicles to unknowingly join in the congestion before the routing tables are updated. To solve this drawback and to consider accidents implicitly without any concern of the two aforementioned points, TT_{ij}^k, D_{ij}^k and LL_{ij}^k are used simultaneously at Eq. (12) in AVCAS. By using this new approach, the ants get an additional penalty if they choose the congested roads or the roads with longer travel time which may happen in an accident situation. In this way, roads with less traffic density and travel time are favored even if those roads are somewhat longer.

Another important issue which should be considered in vehicle traffic routing is that, most of the time, all roads are occupied during rush hours making rerouting impossible. Vehicle traffic routing is effective and applicable as long as the full capacity of the roads is not occupied or congested. This issue is considered in AVCAS by finding and utilizing n alternative paths between various OD pairs simultaneously from the early stage of the routing system. AVCAS periodically, called re-generation period, γ, generates pre-defined number of FANTs, N_a, in order to compute and use alternative paths in its routing procedure. FANTs are located on each node of segments as origin points. Then, they start to explore road map using Eq. (7) and considering the other nodes of their segment as destination points. They find up to n alternative paths (i.e. at least one path and at most n paths) to the other nodes of their segment. After that, BANTs return to origin points from destination points and update the visited links' pheromone value by using Eq. (11). $Intra-ST_{(i)}$ is created and updated by using this information and procedure. $Intra-ST_{(i)}$ includes m smaller tables where m is the number of nodes in segment i. Each of these smaller tables consists of at least m and at most $m \times n$ rows as a possible destination, since AVCAS computes up to n alternative paths between various OD pairs of its own segment. Moreover, these tables consist of 3 columns: destination node, next node and route probability. It means that different paths with various criteria (e.g. distance, capacity, density, travel time and speed) are used for routing vehicles with the same OD pairs. In this way, road capacities are utilized more efficiently. It is worth noting that these n alternative paths are ordered and got priority based on the probability value calculated by Eq. (7) which encompasses all the mentioned criteria for each same OD pair. It means that the path with highest probability value gets the highest priority (n), while the path with lowest probability value gets the lowest priority (1). Vehicles are routed through these alternative paths in such a way that more number of vehicles are routed to the path with the highest probability value and the fewer number of them are routed to the path with the lowest probability value. Cross-multiplication rule is used in Eqs. (13) and (14) in order to identify the portion of routing requests that should be routed through each of the n found alternative paths as follows:

$$Z = \frac{100}{n+(n-1)+\cdots+1}, \tag{13}$$

$$P_n = n \times Z. \tag{14}$$

where P_n indicates the portion or percentage of the vehicles that should be guided through the path with the priority n. Besides, $n, n-1, \ldots, 1$ are the mentioned priorities assigned to the n alternative paths.

This may appear to cause delays for some vehicles because they are routed through longer routes but in fact, it leads to a shorter average travel time for the whole cohort because the congestion is spread to n paths instead of only one path. The optimal number of alternative paths (n), pre-defined number of FANTs (N_a) and re-generation period (γ) are obtained through the simulation results.

4.4. Stopping procedure

ACO should be stopped or completed when a predefined condition(s) is reached. A predefined number of iterations, execution time or maximum visited nodes by ants and the pheromone value remaining unchanged for a number of consecutive iterations are some examples of the stopping criteria. However, AVCAS executes for an infinite number of cycles. A cycle completed by reaching a predefined number of iterations where an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination and is set to $n+1$, where n is the number of nodes in a specific segment. It can also be used as an algorithm loop prevention criteria.

4.5. AVCAS in the real world

To implement AVCAS in the real world, the road map is split into different segments using segmentation phase and one server is assigned to each segment. Each navigation server is responsible for a spatially limited area, where it handles routing requests from vehicles within its segment. Vehicle-to-infrastructure communication is a necessity for collecting real-time traffic information. Every crossing vehicle sends some information such as time, direction and destination to the located road side units at the junctions. This information is then transferred to the navigation servers which are the cornerstone of AVCAS. Vehicle position is transmitted by GPS-enabled devices (e.g. personal navigation assistant or smart phone) to its nearby road side unit. Navigation servers use this information to compute HTS, CTS and PTS for each road within their own segment by using Eqs. (4)–(6), as explained in Fig. 2.

Navigation servers regenerate a number of FANTs (N_a) at predefined TIs, namely re-generation period (γ) and use them to compute up to n alternative paths (i.e. at least one path and at most n paths) between various OD pairs of its own segment by applying AVCAS. After that, BANTs return to origin points from destination points and update the visited links' pheromone value by using Eqs. (11) and (12). $Intra-ST_{(i)}$ is created and updated by using this information and procedure. $Intra-ST_{(i)}$ is used to guide vehicles to their destination if they are within their destination segment, otherwise they are routed to the proper border nodes. Vehicles with a same origin and destination are routed through n calculated alternative paths instead of one path.

The navigation servers communicate via wired networks due to high security and their resistance against interference. They utilize the border nodes and routing layers' (third and fourth layers of our layering model) information to create and update *BNT* and *Inter-ST* in order to find proper border nodes to route the vehicles between two different segments. Using all these data, an OD routing is performed, where each vehicle receives an individual routing guidance based on its current position and destination before each junction in due time by using infrastructure-to-vehicle communication. There is a serious real-time challenge to be solved since each vehicle has its own deadline to receive the routing information based on its speed. AVCAS efficiency is evaluated through simulation in the next section.

5. Simulation results

In this section, the simulation results are discussed by comparing the proposed AVCAS with other existing systems. It is worth

noting that since vehicular environment changes dynamically and quickly based on many unpredictable factors, the choice of an efficient and meaningful simulation setup was a very difficult task. In order to overcome this difficulty, a limited set of tunable factors was defined such as the topological and physical properties of the selected road map, traffic patterns, evaluation metrics, and competing algorithms and their variables' values.

5.1. Simulation setup

First, a part of the city of Kuala Lumpur, Malaysia map was extracted from OpenStreetMap in the form of XML formatted.osm files and used as the physical road map in our simulation which is illustrated in Fig. 4. Different numbers of lanes and speed limits were assigned to different roads in order to get closer to the real vehicular scenarios. Table 1 represents the different statistic specifications for this road map.

SUMO 0.10.0 was used to generate the vehicle traffic and movement patterns on the extracted map of Kuala Lumpur. SUMO (Krajzewicz et al., 2012) is the most widely used open-source and time discrete microscopic road traffic simulation package available. Along with the road map network, some other specifications such as speed limits, number of lanes and detectors at junctions are defined in SUMO. We have utilized Netconvert and Trafficmodeler (Papaleondiou and Dikaiakos, 2009) tools in SUMO to convert the map into a SUMO suitable format (i.e. from osm to net file in XML format) and to generate vehicular traffic and movements. Various numbers of vehicles (i.e. ranging from 100 to 1000) were generated and located on each of the desired origins. In Fig. 4, arrows represent these origins and the highlighted red road is the vehicle's destination. In SUMO, we have utilized the default settings for most of the attributes; however the value of some of these attributes has been changed. Table 2 represents these attributes and their default and new values.

The first 60 s of the output from SUMO was discarded to get more accurate results. TraNSLite, which is a GUI tool for generating realistic mobility traces for simulating vehicular networks in NS-2, was used to convert the generated traffic scenario into a usable format for NS-2.33. The output of TraNSLite is a TCL file which was used as the traffic pattern for NS-2. AVCAS was simulated in NS-2 and its output, i.e. the new values for each link of the map, was transferred to SUMO for evaluation purposes. This process is illustrated in Fig. 5 and Table 3 summarizes the simulation parameters used for NS-2. All results represent an average of over 25 executions with different scenarios (maximum error of 10% with a degree of confidence of 90%). The evaluation process consists of two steps: (1) finding the proper values for AVCAS parameters using NS-2 and (2) comparing AVCAS with other mechanisms using SUMO. A static shortest path algorithm (e.g. Dijkstra algorithm, Dijkstra, 1959) was used as the default algorithm in SUMO. Therefore, we used it with the new link value in order to compare our algorithm to other existing algorithms (Papaleondiou and Dikaiakos, 2009).

5.2. Simulation results for AVCAS parameters' value

To find the best value for the various parameters of AVCAS, their impact on this algorithm were examined individually

Table 1
Statistic specifications of road map.

Specification	Value
Dimension	4 km × 3 km
Map area	12 km^2
Streets/km^2	240.25
Junctions/km^2	150.3
Avg. street length	205.5 m
Avg. lanes/street	1.9

Fig. 4. Physical road map of selected part of Kuala Lumpur city. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 2
Default and new values for some attributes of SUMO.

Attribute	Default value	New value
Acceleration	2.6 m/s^2	0.5, 1.5, 2.5 m/s^2
Deceleration	4.5 m/s^2	1, 2, 3 m/s^2
Sigma (driver imperfection)	0.5	0, 0.5, 1
Minimum gap	2.5 m	2 m
Maximum speed	70 m/s	30 m/s

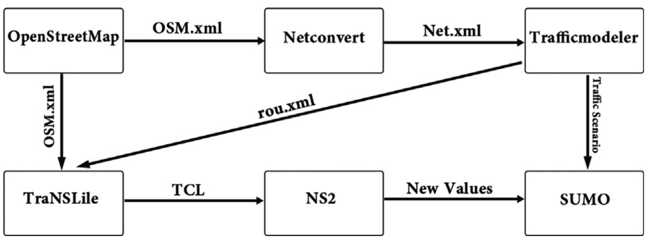

Fig. 5. The simulation procedure of the proposed AVCAS.

Table 3
Configuration parameters in the simulation.

Parameter	Value
Simulation time	1000 s
Size of messages	500 bytes
Vehicle speed	0–30 m/s
MAC/PHY	IEEE 802.11p
Vehicles density	100–1000
Mobility generator	SUMO
Max. transmission range	400 m

through simulation and explained as follows. A simple road map, which includes 12 nodes (junctions) of which 2 are border nodes, 15 links (roads), and 2 segments, was used for this purpose and is illustrated in Fig. 6. The average travel time was utilized as a measurement criterion in this section.

HTS information power (ξ): This parameter specifies the effect of HTS information on the short-term prediction travel speed of roads. The historical data influence has increased by raising the value of ξ, while, decreasing the value of ξ will reduce the effect of historical data on the path selection procedure.

CTS information power (λ): The function of this parameter is very similar to ξ but the difference is that it controls the CTS information impact on the path selection procedure. It is worth noting that this information is gathered by VANTs.

Therefore, there should be a proper trade-off between ξ and λ (i.e., $\xi+\lambda=1$, called weighted mean, Terr, 2004). In our simulation environment, the best condition occurs when $\xi=0.4$ and $\lambda=0.6$ for AVCAS evaluation. Fig. 7 illustrates the average travel time of the found paths by AVCAS as a function of the HTS and the CTS information power, considering other parameters as follows: $\alpha=0.5$, $\rho=0.5$, $N_a=15$, $n=3$, $\gamma=5$ TIs (50 s). The average travel time converges towards two different values at the beginning (λ from 0 to 0.2) and at the end (λ from 0.8 to 1) of this diagram. This is because, at the beginning, path finding is more based on HTS information ($0.8 \leq \xi \leq 1$) whereas at the end, it is more based on CTS information ($0.8 \leq \lambda \leq 1$). Our obtained results, assigning higher value to λ compared to ξ (i.e. $\lambda=0.6$, $\xi=0.4$), can be supported by the following reasons: (1) considering non-recurring congestion condition (i.e. accident, working zones, and weather conditions) in vehicle routing is one of our main concerns. Based on the results obtained by Rakha and Van Aerde (1995), the traffic conditions vary considerably from one day to the next day

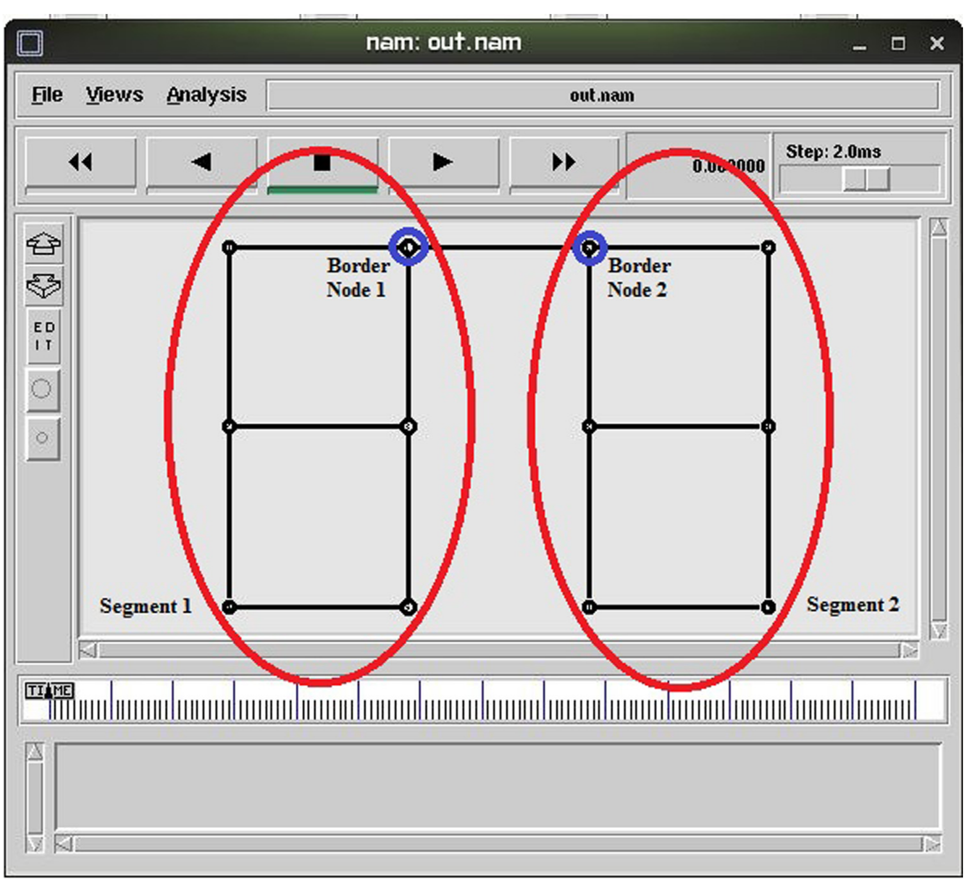

Fig. 6. Map used for finding AVCAS parameters' values.

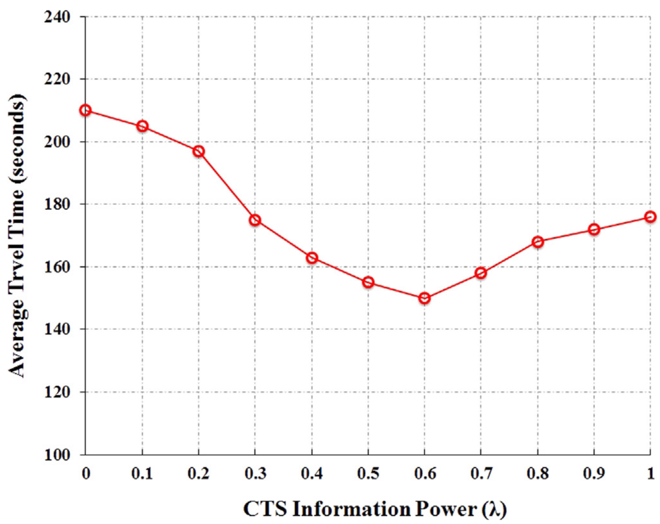

Fig. 7. Average travel time for AVCAS as a function of CTS information power (α=0.5, β=0.5, ρ=0.5, N_a=15, n=3, γ=5 TIs (50 s)).

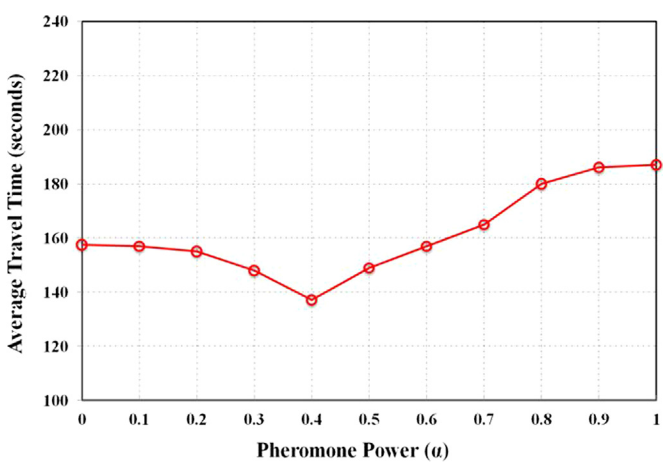

Fig. 8. Average travel time for AVCAS as a function of pheromone power (λ=0.6, ξ=0.4, ρ=0.5, N_a=15, n=3, γ=5 TIs (50 s)).

in non-recurring congestion condition. Consequently, the historical data will be insufficient for commuters to find the optimum routes through the network, and the provision of current traffic information could provide major benefits, and (2) moreover, theoretically and based on the obtained results by Karbassi and Barth (2003), the prediction gets more and more close to the true value with increasing the number of real-time observations.

Pheromone power (α): This parameter specifies the probability of a link to being selected based on its pheromone value and also the impact of the gathered data by PANTs. The data influence and the exploitative nature of PANTs are increased by increasing the value of α. Decreasing the value of α will increase the PANTs' exploration and decrease the effect of pheromone value on the path selection procedure.

Real-time information power (β): The function of this parameter is very similar to α but the difference is that it controls the real-time information impact on the path selection procedure. This information is gathered by VANTs.

Similar to ξ and λ, there should be a proper trade-off between α and β (i.e., $\alpha + \beta = 1$). The best condition occurs when α=0.4 and β=0.6 in our simulation environment for AVCAS evaluation. Fig. 8 illustrates the average travel time of the found paths by AVCAS as a function of the pheromone and real-time information. The average travel time converges towards two different values: 160 s and 185 s at the beginning (α from 0 to 0.2) and at the end (α from 0.8 to (1) of this diagram, respectively. This is because, at the beginning, path finding is based more on vehicles real-time information ($0.8 \le \beta \le 1$) whereas at the end, it is based more on pheromone trial information ($0.8 \le \alpha \le 1$).

Pheromone evaporation rate (ρ): Based on Di Caro (2004), this parameter plays an important role when there are multiple paths for selection and the characteristics of the environment change rapidly and dynamically. The described status is very similar to the vehicular environment which is the main focus of this paper. Since ρ has a direct effect on having the proper trade-off between exploration and exploitation as well as the convergence speed of the algorithm, different values are examined for finding the best value of this parameter through the simulation and its result is demonstrated in Fig. 9. At low values of ρ, the convergence speed is high because of the slow changes in the pheromone value of the links, while the algorithm does not converge at higher values of ρ because of the quick changes of pheromone trails on the links. The lowest average travel time happened when ρ=0.3.

Number of alternative paths (n): It is worth noting that although a large number of alternative paths for each OD pair allow better vehicle congestion avoidance and balancing, this leads to higher computational complexity. Moreover, since distance is one of the main metrics in AVCAS, a large number of alternative paths which leads to computational overhead and long paths are not necessary. Selecting the proper value for this parameter can lead to decreasing

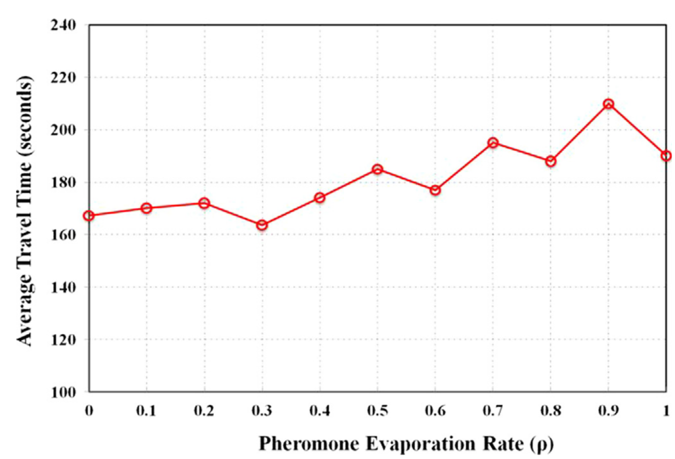

Fig. 9. Average travel time for AVCAS as a function of pheromone evaporation rate (λ=0.6, ξ=0.4, α=0.4, β=0.6, N_a=15, n=3, γ=5 TIs (50 s)).

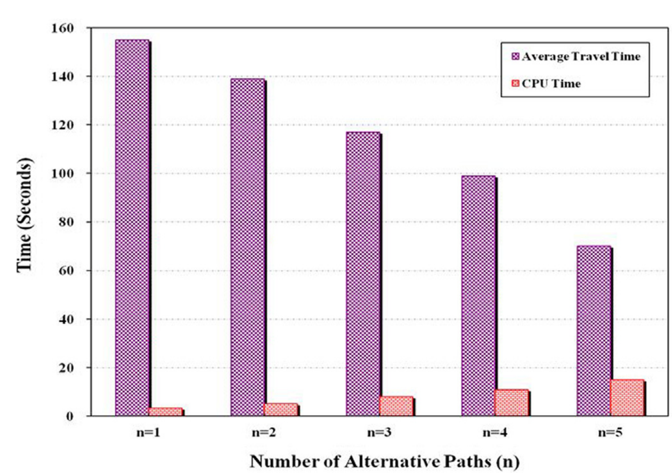

Fig. 10. Average travel time and CPU usage time for AVCAS as a function of number of alternative paths (λ=0.6, ξ=0.4, α=0.4, β=0.6, ρ=0.3, N_a=15, γ=5 TIs (50 s)).

both the average travel time and the computational cost. Since system response time is very critical criteria in a vehicular environments because of rapid changes, $n=3$ was selected for AVCAS based on the results in Fig. 10.

It means that AVCAS finds up to 3 alternative paths for each OD pairs in each segment. These three alternative paths are ordered based on the obtained probability value by Eq. (7) which encompasses various criteria (e.g. distance, capacity, density, travel time and speed) for each same OD pair. Consequently, the path with highest probability value has higher priority and a chance of being suggested to vehicles. For each OD pair, a First Come First Serve (FCFS) strategy is used in order to route the vehicles through these three alternative paths. The first 50% of routing requests are routed via the first path, i.e. the least congested shortest path. The next 30% of routing requests are routed through the second path which is longer than the first path but is still less congested. The last 20% of routing requests are routed via the last path (3rd path). This routing cycle is continued for the coming routing requests. If AVCAS finds at most 2 alternative paths between specific OD pair, it routes 70% of vehicles through the first path and the other 30% is routed via the second path. However, all the vehicles are routed via the same path if there is only one path between specific OD pair. This last case is usually occurred when the vehicles are close to their destination. All of the above-mentioned percentages are obtained via Eqs. (13) and (14).

Number of ants (N_a) *and re-generation period* (γ): In AVCAS, the new path finding process is started periodically by regenerating a predefined number of new FANTs at predefined TIs. In general, a lower value for γ and a higher value for N_a lead to better average travel time and algorithm convergence speed, respectively. As a result, the computational cost and communication overhead of the system have increased. Fig. 11 illustrates the average travel time for different values of γ.

Considering the trade-off between the average travel time on one side and the communication overhead and computational cost on the other, 30 s or 3 TIs is selected as the regeneration period of FANTs in AVCAS. Moreover, the number of ants is chosen as a function of the number of destination nodes ($n-1$) in the segment and alternative paths (the number of nodes (source) in the segment $\times$ alternative paths). In our scenario, 15 ants (i.e. 3 (alternative paths) $\times$ 5 (destination nodes in the segment)) are put at each of the start point (source). The configuration parameters of AVCAS in NS-2 are summarized in Table 4.

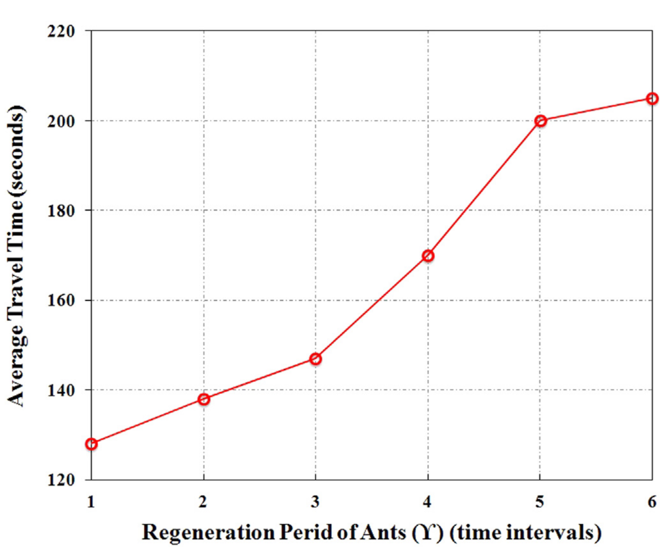

Fig. 11. Average travel time for AVCAS as a function of the regeneration period of ants (λ=0.6, ξ=0.4, α=0.4, β=0.6, ρ=0.3, n=3).

Table 4
Configuration parameters of AVCAS in NS-2.

Parameter	Examined range	Proper value
ξ	0–1 (step: 0.1)	0.4
λ	0–1 (step: 0.1)	0.6
α	0–1 (step: 0.1)	0.4
β	0–1 (step: 0.1)	0.6
ρ	0–1 (step: 0.1)	0.3
γ	1 TI–6 TI (step: 1 TI)	3 TIs or 30 s
n	1–5 (step: 1)	3

5.3. Simulation results for AVCAS evaluation

After investigating the effects of different values on the AVCAS parameters and finding their proper values, the efficiency of AVCAS was evaluated by comparing with Dijkstra (1959), pure ant colony optimization (PACO) (Fan et al., 2004) and HRS (Tatomir and Rothkrantz, 2006) systems. The Dijkstra algorithm was selected because it is a very popular algorithm and is used in most of the CNSs for finding the shortest paths. The ACO algorithm was utilized in the AVCAS mechanism and PACO was chosen to verify and validate our proposed changes and modifications. The comparison between PACO and AVCAS is summarized in Table 5. Finally, the HRS algorithm was compared with AVCAS because of their similarities in the use of segmentation and an ant-based algorithm for path finding and vehicle congestion reduction. These four mechanisms were compared, based on the average travel time, speed and distance by considering various vehicle densities ranging from 100 to 1000 vehicles. These three metrics were selected due to their predominant role in vehicular environments.

Average travel time: This metric was calculated and the results are illustrated in Fig. 12 for each of the mentioned mechanisms (i.e. Dijkstra, PACO, HRS and AVCAS). The obtained results, which are confirmed by Daganzo (1994), show that a vehicle's average travel time has a direct relationship with vehicle density: as the number of vehicles increases, the average travel time increases. However, this increment was too sharp in the Dijkstra system versus other systems. This is because by using the Dijkstra system; all of the vehicles with the same OD were guided through the same path without paying attention to other factors such as vehicle congestion and accidents. In comparison, ant-based systems (i.e. PACO, HRS and AVCAS) improved the average travel time significantly. The average travel time at low vehicle densities (from 100 to 400) was almost the same for all systems, while this metric varied between each system at higher densities (from 500 to 1000). AVCAS had the best results in different vehicle densities and it decreased travel time up to 19%, 11% and 6% compared with Dijkstra, PACO and HRS, respectively. AVCAS improves the average travel time since it avoids congestion instead of recovering from it, which is not considered in PACO and HRS. Moreover, in PACO and HRS, if many vehicles have the same OD pair at the same time, congestion can be transferred from one road to another. This problem is solved in AVCAS since it balances the traffic flow using up to three alternative paths (n).

Average travel speed: The relationship between the average travel speed and CO_2 emissions (and therefore, fuel consumption) was investigated by Barth and Boriboonsomsin (2009) and they discovered that there is a U-shape relationship between these metrics. This means that at a very low average travel speed, which normally occurs during vehicle congestion, and with a high number of stop-and-go driving events and extended engine idling on the road, the fuel consumption as well as CO_2 emissions increased by an average of 30% (Barth and Boriboonsomsin, 2009; Spalding, 2008). Conversely, at very high speeds, the vehicle's engine requires more power which leads to higher fuel

consumption and more CO_2 emissions. Moderate average travel speeds ranging from 40 to 60 mph ($\simeq$ 17.9–26.8 m/s) leads to the lowest fuel consumption and CO_2 emissions. The average travel speed of Dijkstra, PACO, HRS and AVCAS at various vehicle densities is illustrated in Fig. 13. AVCAS obtained the best average speed rate at all vehicle densities by avoiding congestion and providing alternative paths before congestion occurred. By increasing the vehicle density, the average speed decreased smoothly from 25 to 17.7 m/s in AVCAS and this speed range is the reported range for low fuel consumption and CO_2 omissions as in Barth and Boriboonsomsin (2009). The worst average travel speed and average travel time were generated by the Dijkstra algorithm and ranged from 25 to 10.5 m/s. The results for low vehicle densities (e.g. 100, 200 vehicles) were the same for all systems since the congestion level is low and all systems route the vehicles via the same path which is the shortest path.

Average travel distance: This metric was calculated and is illustrated in Fig. 14 for Dijkstra, PACO, HRS and AVCAS. The worst average travel was associated with AVCAS and this is related to having a better average travel speed and time compared with the other systems. AVCAS improves congestion by proposing slightly longer paths with less congestion instead of the shortest paths with congestion. This leads to an increase in the travel distance but a decrease in the travel time and speed. It is worth noting that the average travel distance had increased at most 15% compared to the shortest path which was proposed by Dijkstra. Since AVCAS utilizes three alternative paths for avoiding and reducing vehicle congestion, its average travel distance is higher than PACO and HRS which propose one alternative path and may transfer the congestion from one point to another point. Moreover, the average travel distance is constant and not dependent on vehicle density, congestion or accidents and therefore is less suitable for vehicle congestion and avoidance systems.

Usage rate: It cannot be assumed that in a real world scenario, every driver will follow the routing systems guidance. Therefore, in addition to the average travel time, speed and distance, the usage rate (i.e. the proportion of drivers who use a specific guidance system and accept its guidance) was also investigated in this paper. Fig. 15 presents the effect of various usage rates of Dijkstra, PACO, HRS and AVCAS on an average travel time. The results indicate that the average travel time can be enhanced almost at the same rate using all of the mentioned systems under low usage rate (from 10% to 40%). The lowest average travel time for Dijkstra, PACO and HRS was achieved when half of the drivers used these systems. If the usage rate of these systems exceeds 50% of the drivers, it has a negative impact on the average travel time. This is because, most of the vehicles are routed to the same route thus congestion is increased. However, periodic re-routing and considering link (road) travel time reduce this negative impact on PACO and HRS, respectively. AVCAS had the best result in this case and improved the average travel time even at higher usage rates (from 50% to 80%), however its performance degraded slightly for very high usage rates (from 90% to 100%) due to the higher than necessary re-routing for congestion avoidance.

Reaction for accident: This scenario was simulated to evaluate the behavior of AVCAS when an accident takes place compared with the other three systems. We split the vehicles into four categories based on their chosen routing system, i.e. Dijkstra, PACO, HRS and AVCAS. We generated 100 vehicles for each category and dedicated 25 vehicles of each category to four starting points (i.e. arrows in Fig. 4) in our simulation environment. The total simulation period was 1000 s. An accident was generated at one of the main roads, which is depicted via a cross sign in Fig. 4, after 300 s. This accident was omitted from the road at the 700th second of simulation. Vehicles may be forced to stop or halt on the lane for a defined time span by using the stop element in SUMO and this works similar to real accidents on the roads (Hrizi and Filali, 2010). The travel time necessary for vehicles to reach their destination was counted for each category to measure the performance of these systems and is illustrated in Fig. 16. As it can be seen before the accident happened, all of the vehicles are guided through the shortest path via Dijkstra, PACO and HRS, and through the n alternative least congested shortest paths via AVCAS. However, after the accident and because of the congestion on the link with the accident, PACO and HRS started to reroute the vehicles through another alternative path, while AVCAS continued vehicles routing through the n alternative least congested shortest paths. This did not happen in Dijkstra due to the lack of attention to the dynamic changes of the vehicular environments. As a result, travel time started to increase sharply for vehicles routed via the Dijkstra system. In addition, travel time started to increase earlier for vehicles routed via HRS and AVCAS due to the use of prediction in these two systems. This increment for HRS was greater than AVCAS since HRS blocks the link with the accident while AVCAS reduces its chosen probability. From 400 to 700 s of simulation, PACO, HRS, AVCAS rerouted the vehicles through the alternative paths and stabilized the travel time value. At the 700th second when the accident was cleared from the link, the average travel time decreased rapidly for all of the systems and all the graphs smooth out to reach their initial values. AVCAS had the best reaction for congestion since it uses travel time and vehicle density and travel speed prediction for vehicle routing and uses alternative paths from the beginning before congestion happens.

6. Conclusion

This paper addresses vehicle traffic congestion, which is one of the major challenges of metropolises. Congestion is affected by the limited capacity of roads and the high number of vehicles on the road. An ant-based algorithm was combined with map segmentation and the average travel speed prediction of roads in order to derive an improved congestion avoidance system. Segmentation and short-term prediction were used to overcome the dynamicity and quick changes of vehicular environments. Applying an ant-based algorithm to our system required some modifications to the basic and original ACO algorithm. These modifications include map segmentation and layering, new probability function, and new reinforcement and evaporation rules and parameters.

The NS-2 simulator was utilized to find the best values for AVCAS parameters (i.e., $\alpha, \beta, \rho, \gamma$ and n). The experimental results show how the performance of AVCAS can be changed by different values for the parameters. Using these modifications makes AVCAS more easily deployable in distributed and dynamic vehicular environments. AVCAS's efficiency was evaluated by comparing it with other algorithms such as Dijkstra, PACO and HRS, taking into consideration the average travel time, speed and distance as the evaluation metrics. The results from SUMO show that AVCAS outperforms the others in the case of the average travel time and speed even at high usage rates (i.e., $\geq 50\%$).

Further studies might focus on designing a smarter system which considers different types of vehicles such as ambulances, police and fire trucks in its guidance system. It would also be useful to develop a more cost-effective system that uses vehicle-to-vehicle communications instead of vehicle-to-infrastructure communications, which are used in AVCAS. Besides, other features of computing intelligence such as adaptation, flexibility and learning will be considered as an extension of AVCAS. Value of time (VOT) represents how much money the user is willing to trade off for time saving. By considering this preference as an input

Table 5
Comparison between PACO and AVCAS based on their steps.

Step	PACO (Fan et al., 2004)	AVCAS
Problem graph depiction	The problem graph is changed to a tree graph	Segmentation and layering phases are used for problem graph preparation (see Section 4.1)
Initialization	A number of FANTs (m) are located on the nodes and started to explore the problem graph by choosing the next node using probability function as follows: $P_{ij} = \frac{(\tau_{ij}) - \eta_{ij}^{\beta}}{\sum_{r \in A_k} \tau_{ir} \times \eta_{ir}^{\beta}},$ where $\eta_{ij} = 1/d_{ij}, d_{ij}$ is the distance between nodes i and j. A_k is the reachable node set of ant k at node i	Two types of ant are defined, namely VANT and PANT, to consider historical and current traffic conditions and also routing tablesâ€™update. PANTs are further divided into two FANTs and BANTs. A number of FANTs (N_a) are located on the nodes and started to explore the problem graph by choosing the next node using probability function as follows: $P_{ij} = \frac{\alpha(\tau_{ij}) + \beta(\eta_{ij})}{\sum_{h \in tabu_k} (\alpha(\tau_{ij}) + \beta(\eta_{ij}))} \times \left(\frac{1}{1 + \frac{1}{N_j}} \right),$ $\eta_{ij} = (1 - D_{ij}) + \left(\frac{PTS_{ij}}{\varphi} \right),$ Refer to Section 4.2 for more details about the above-mentioned equations and their variables
Pheromone update	The pheromone intensity is updated via BANTs while returning to the source nodes from destination nodes by using pheromone update rule as follows: $\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \rho(\Delta\tau_{ij}),$ $\Delta\tau_{ij} = \begin{cases} \frac{1}{L} & \text{if link } (i,j) \text{ traversed}, \\ 0 & \text{otherwise}. \end{cases}$ where $\rho \in (0, 1]$ and L are pheromone evaporation ratio and average length of cycle, respectively	The pheromone intensity is updated via BANTs while returning to the source nodes from destination nodes by using pheromone update rule as follows: $\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k},$ $\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{TT_{ij}^{k}} + \frac{1}{D_{ij}^{k}} + \frac{1}{LL_{ij}^{k}} & \text{if link } (i,j) \text{ traversed}, \\ 0 & \text{otherwise}. \end{cases}$ Refer to Section 4.3 for more details about mentioned equations and their variables
Stopping procedure	PACO is completed by reaching a predefined number of iterations, T	AVCAS executes for an infinite number of cycles. A cycle completed by reaching a predefined number of iterations where as an ant is dropped by arriving at a predefined maximum number of hops before reaching its destination and is set to $n+1$, where n is the number of nodes in a specific segment

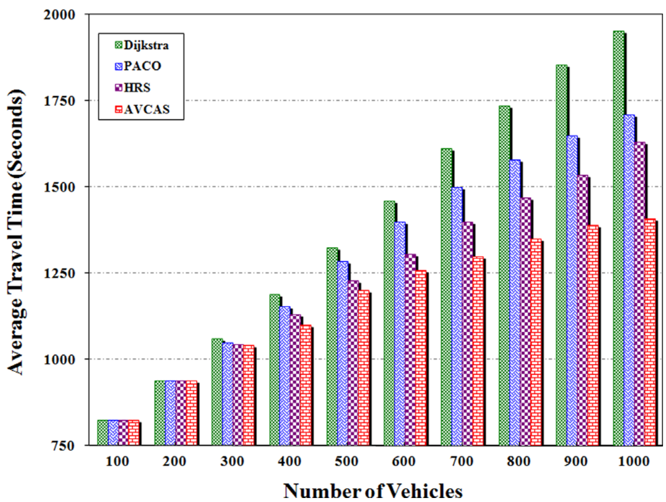

Fig. 12. Average travel time for Dijkstra, PACO, HRS and AVCAS as a function of vehicle density.

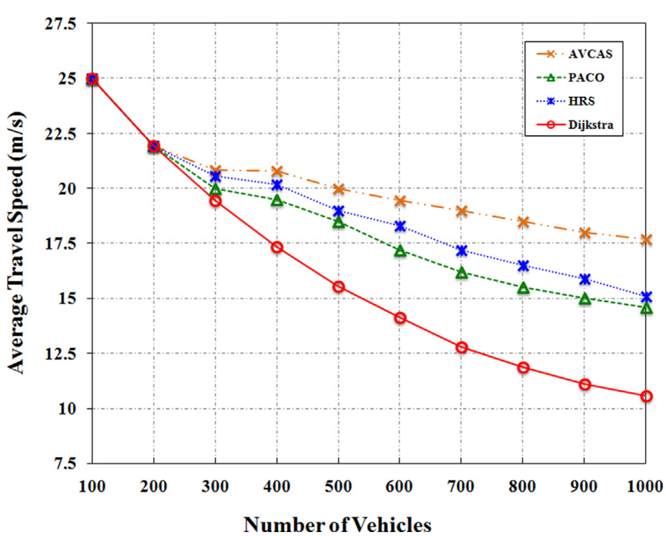

Fig. 13. Average travel speed for Dijkstra, PACO, HRS and AVCAS as a function of vehicle density.

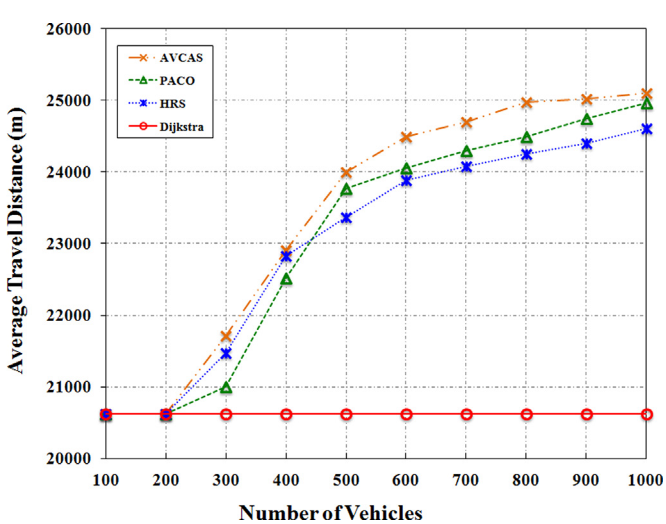

Fig. 14. Average travel distance for Dijkstra, PACO, HRS and AVCAS as a function of vehicle density.

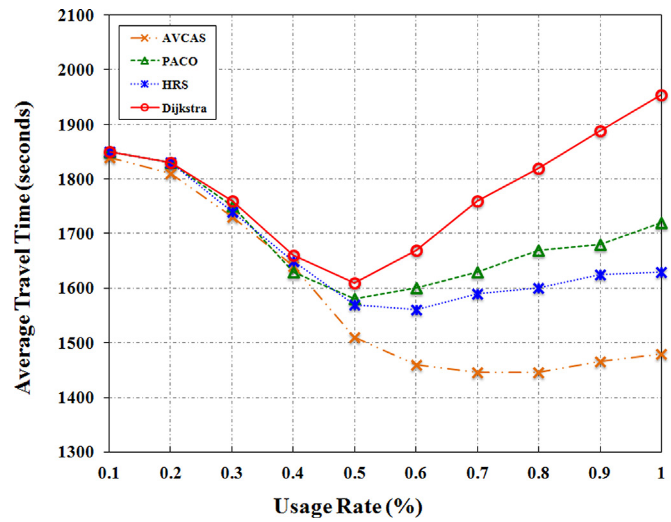

Fig. 15. Average travel time for Dijkstra, PACO, HRS and AVCAS as a function of their usage rate.

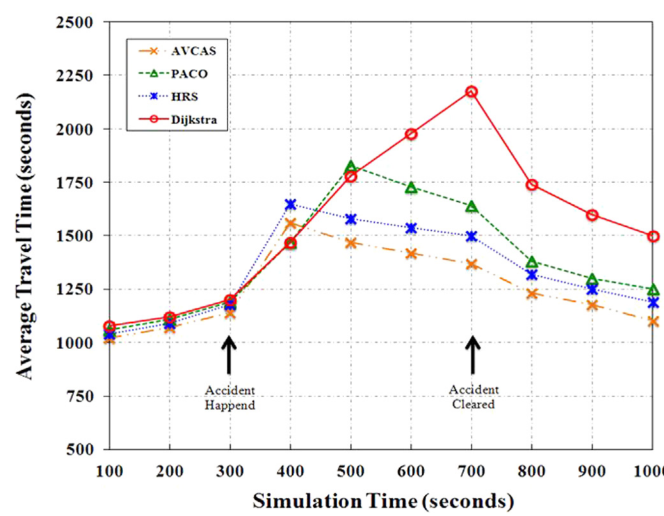

Fig. 16. Average travel time for vehicles that used Dijkstra, PACO, HRS and AVCAS as a routing system as a function of simulation time.

variable for AVCAS, we can contribute to the improvement of model in our future work.

Acknowledgments

This research is supported by UM High Impact Research MoE Grant UM.C /625/1/HIR/MOHE/FCSIT/09 from the Ministry of Education Malaysia.

References

Abbas, S., Khan, M.S., Ahmed, K., Abdullah, M., Farooq, U., 2011. Bio-inspired neuro-fuzzy based dynamic route selection to avoid traffic congestion. Int. J. Sci. Eng. Res. 2.

Ahmed, H., Glasgow, J., 2012. Swarm Intelligence: Concepts, Models and Applications. Technical Report. School of Computing, Queens University.

Ando, Y., Masutani, O., Sasaki, H., Iwasaki, H., Fukazawa, Y., Honiden, S., 2006. Pheromone model: Application to traffic congestion prediction. In: Engineering Self-Organising Systems. Springer, Germany. pp. 182–196.

Barth, M., Boriboonsomsin, K., 2009. Traffic Congestion and Greenhouse Gases. The Magazine of University of California Transportation Center (UCTC).

Batool, F., Khan, S.A., 2005. Traffic estimation and real time prediction using adhoc networks. In: Proceedings of the IEEE Symposium on Emerging Technologies, IEEE. NJ, USA. pp. 264–269.

Bedi, P., Mediratta, N., Dhand, S., Sharma, R., Singhal, A., 2007. Avoiding traffic jam using ant colony optimization-a novel approach. In: International Conference on Conference on Computational Intelligence and Multimedia Applications, IEEE. NJ, USA. pp. 61–67.

Berrichi, A., Yalaoui, F., Amodeo, L., Mezghiche, M., 2010. Bi-objective ant colony optimization approach to optimize production and maintenance scheduling. Comput. Oper. Res. 37, 1584–1596.

Blum, C., 2005. Ant colony optimization: introduction and recent trends. Phys. Life Rev. 2, 353–373.

Bonebeau, E., Dorigo, M., Theraulaz, G., 1999. Swarm Intelligence: From Natural to Artificial Systems, vol. 222. Oxford Press, pp. 223–248.

Boryczka, M., Bura, W., 2013. Ant colony optimization for the multi-criteria vehicle navigation problem. In: Agent-Based Optimization. Springer, Berlin, Heidelberg. pp. 29–53.

Bullnheimer, B., Hartl, R.F., Strauss, C., 1997. A New Rank Based Version of the Ant System. A Computational Study.

Chandra Mohan, B., Baskaran, R., 2012. A survey: ant colony optimization based recent research and implementation on several engineering domain. Expert Syst. Appl. 39, 4618–4627.

Chen, W.N., Zhang, J., Chung, H.H., Huang, R.Z., Liu, O., 2010. Optimizing discounted cash flows in project scheduling an ant colony optimization approach. IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 40, 64–77.

Cheung, S.Y., Coleri, S., Dundar, B., Ganesh, S., Tan, C.W., Varaiya, P., 2005. Traffic measurement and vehicle classification with single magnetic sensor. Transp. Res. Rec.: J. Transp. Res. Board 1917, 173–181.

Chien, S.I.J., Kuchipudi, C.M., 2003. Dynamic travel time prediction with real-time and historic data. J. Transp. Eng. 129, 608–616.

Claes, R., Holvoet, T., 2011. Ant colony optimization applied to route planning using link travel time predictions. In: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW). IEEE. NJ, USA. pp. 358–365.

Claes, R., Holvoet, T., 2012. Cooperative ant colony optimization in traffic route calculations. In: Advances on Practical Applications of Agents and Multi-Agent Systems. Springer, Berlin, Heidelberg. pp. 23–34.

Coifman, B.A., Mallika, R., 2007. Distributed surveillance on freeways emphasizing incident detection and verification. Transp. Res. Part A: Policy Pract. 41, 750–767.

Daganzo, C.F., 1994. The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. Part B: Methodol. 28, 269–287.

Deneubourg, J.L., Aron, S., Goss, S., Pasteels, J.M., 1990. The self-organizing exploratory pattern of the argentine ant. J. Insect Behav. 3, 159–168.

Desai, P., Loke, S.W., Desai, A., Singh, J., 2013. Caravan: Congestion Avoidance and Route Allocation using Virtual Agent Negotiation.

Dhillon, S., Van Mieghem, P., 2007. Performance analysis of the antnet algorithm. Comput. Netw. 51, 2104–2125.

Di Caro, G., 2004. Ant Colony Optimization and its Application to Adaptive Routing in Telecommunication Networks (doctoral dissertation).

Di Caro, G., Dorigo, M., 2011. Antnet: Distributed Stigmergetic Control for Communications Networks. arXiv preprint arxiv:1105.5449.

Dia, H., 2001. An object-oriented neural network approach to short-term traffic forecasting. Eur. J. Oper. Res. 131, 253–261.

Dias, J.A.C., Machado, P., Pereira, F.C., 2013. Privacy-aware Ant Colony Optimization Algorithm for Real Time Route Planning.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

Dimitrakopoulos, G., Demestichas, P., 2010. Intelligent transportation systems. IEEE Veh. Technol. Mag. 5, 77–84.

Dooms, A., 2013. Parallel Multi-Agent Path Planning in Dynamic Environments for Real-time Applications.

Dorigo, M., 1992. Optimization, Learning and Natural Algorithms (Ph.D. thesis). Politecnico di Milano, Italy.

Dorigo, M., Birattari, M., 2010. Ant colony optimization. In: Encyclopedia of Machine Learning. Springer, Berlin, Heidelberg. pp. 36–39.

Dorigo, M., Di Caro, G., Gambardella, L.M., 1999. Ant algorithms for discrete optimization. Artif. life 5, 137–172.

Dorigo, M., Maniezzo, V., Colorni, A., Maniezzo, V., 1991. Positive Feedback as a Search Strategy. Technical Report.

Dorigo, M., Stützle, T., 2004. Ant Colony Optimization. MIT Press, pp. 223–260 (Chapter 6), Cambridge, Massachusetts, USA.

Dorigo, M., Stützle, T., 2010. Ant colony optimization: overview and recent advances. In: Handbook of Metaheuristics. Springer, Berlin, Heidelberg. pp. 227–263.

Fan, H., Hua, Z., Li, J.J., Yuan, D., 2004. Solving a shortest path problem by ant algorithm. In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, 2004. IEEE. pp. 3174–3177, NJ, USA.

Ge, Y., Wang, J., Wang, G., Jiang, F., 2011. Urban vehicle routing research based on ant colony algorithm and traffic rule restriction. In: High Performance Networking, Computing, and Communication Systems. Berlin, Heidelberg, Springer. pp. 95–102.

Ghazy, A.M., EL-Licy, F., Hefny, H.A., 2012. Threshold based antnet algorithm for dynamic traffic routing of road networks. Egypt. Inf. J. 13, 111–121.

Gong, Q., Li, Y., Peng, Z.R., 2008. Trip-based optimal power management of plug-in hybrid electric vehicles. IEEE Trans. Veh. Technol. 57, 3393–3401.

Hallam, N., Hartley, M., Blanchfield, P., Kendall, G., 2004. Optimisation in a road traffic system using collaborative search. In: 2004 IEEE International Conference on Systems, Man and Cybernetics. IEEE. NJ, USA. pp. 2008–2012.

Hansen, P., 1980. Bicriterion path problems. In: Multiple Criteria Decision Making Theory and Application. Springer, Berlin, Heidelberg. pp. 109–127.

Hart, P.E., Nilsson, N.J., Raphael, B., 1968. A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. 4, 100–107.

Hrizi, F., Filali, F., 2010. simits: an integrated and realistic simulation platform for vehicular networks. In: Proceedings of the 6th International Wireless Communications and Mobile Computing Conference. ACM. New York City, USA. pp. 32–36.

Jabbarpour, M.R., Malakooti, H., Noor, R.M., Anuar, N.B., Khamis, N., 2014. Ant colony optimisation for vehicle traffic systems: applications and challenges. Int. J. Bio-Inspired Comput. 6, 32–56.

Jiang, Y., Wang, W.L., Zhao, Y.W., 2007. Solving the shortest path problem in vehicle navigation system by ant colony algorithm. In: Proceedings of the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, World Scientific and Engineering Academy and Society (WSEAS). pp. 188–192.

Kammoun, H.M., Kallel, I., Alimi, A.M., Casillas, J., 2010. An adaptive vehicle guidance system instigated from ant colony behavior. In: 2010 IEEE International Conference on Systems Man and Cybernetics (SMC). IEEE. NJ, USA. pp. 2948–2955.

Kanamori, R., Takahashi, J., Ito, T., 2012. Evaluation of anticipatory stigmergy strategies for traffic management. In: VNC, pp. 33–39.

Karbassi, A., Barth, M., 2003. Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. In: IEEE Proceedings of Intelligent Vehicles Symposium, IEEE. NJ, USA. pp. 511–516.

Köhler, E., Langkau, K., Skutella, M., 2002. Time-expanded graphs for flow-dependent transit times. In: Algorithms ESA 2002. Berlin, Heidelberg, Springer. pp. 599–611.

Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L., 2012. Recent development and applications of SUMO—Simulation of Urban MObility. Int. J. Adv. Syst. Meas. 5, 128–138.

Kwon, J., Petty, K., 2005. Travel time prediction algorithm scalable to freeway networks with many nodes with arbitrary travel routes. Transp. Res. Rec.: J. Transp. Res. Board 1935, 147–153.

LaValle, S.M., 2006. Planning Algorithms. Cambridge University Press, Cambridge, United Kingdom.

Liu, J., Fang, Y., Liu, Y., 2007. Ant colony system algorithm for path routing of urban traffic vehicles. In: 2007 IEEE International Conference on Automation and Logistics. IEEE. NJ, USA. pp. 1902–1907.

Martínez, F.J., González-Vidosa, F., Hospitaler, A., Alcalá, J., 2011. Design of tall bridge piers by ant colony optimization. Eng. Struct. 33, 2320–2329.

Martínez, F.J., González-Vidosa, F., Hospitaler, A., Yepes, V., 2010. Heuristic optimization of RC bridge piers with rectangular hollow sections. Comput. Struct. 88, 375–386.

Merkle, D., Blum, C., 2008. Swarm Intelligence: Introduction and Applications.

Misra, S., Dhurandher, S.K., Obaidat, M.S., Verma, K., Gupta, P., 2010. A low-overhead fault-tolerant routing algorithm for mobile ad hoc networks: a scheme and its simulation analysis. Simul. Model. Pract. Theory 18, 637–649.

Nahar, S.A.A., Hashim, F.H., 2011. Modelling and analysis of an efficient traffic network using ant colony optimization algorithm. In: 2011 Third International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN). IEEE. NJ, USA. pp. 32–36.

Nanthawichit, C., Nakatsuji, T., Suzuki, H., 2003. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. Transp. Res. Rec.: J. Transp. Res. Board 1855, 49–59.

Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D., Hortner, H., 2010. Self-organising congestion evasion strategies using ant-based pheromones. IET Intell. Transp. Syst. 4, 93–102.

Nazari, S., Meybodi, M.R., Salehigh, M.A., et al., 2008. An advanced algorithm for finding shortest path in car navigation system. In: First International Conference on Intelligent Networks and Intelligent Systems, 2008. ICINIS'08. IEEE. NJ, USA. pp. 671–674.

Noto, M., Sato, H., 2000. A method for the shortest path search by extended Dijkstra algorithm. In: 2000 IEEE International Conference on Systems, Man, and Cybernetics. IEEE. NJ, USA. pp. 2316–2320.

Ok, S.H., Seo, W.J., Ahn, J.H., Kang, S., Moon, B., 2011. An ant colony optimization approach for the preference-based shortest path search. J. Chin. Inst. Eng. 34, 181–196.

Ong, H., Mahlia, T., Masjuki, H., 2011. A review on emissions and mitigation strategies for road transport in malaysia. Renew. Sustain. Energy Rev. 15, 3516–3522.

Pan, J., Khan, M.A., Sandu Popa, I., Zeitouni, K., Borcea, C., 2012. Proactive vehicle re-routing strategies for congestion avoidance. In: 2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS). IEEE. NJ, USA. pp. 265–272.

Panigrahi, B.K., Shi, Y., Lim, M.H., 2011. Handbook of Swarm Intelligence. Springer, Berlin, Heidelberg.

Papaleondiou, L.G., Dikaiakos, M.D., 2009. Trafficmodeler: A graphical tool for programming microscopic traffic simulators through high-level abstractions. In: IEEE 69th Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE. pp. 1–5.

Park, J., Li, D., Murphey, Y.L., Kristinsson, J., McGee, R., Kuang, M., Phillips, T., 2011. Real time vehicle speed prediction using a neural network traffic model. In: The 2011 International Joint Conference on Neural Networks (IJCNN). IEEE. NJ, USA. pp. 2991–2996.

Rakha, H., Van Aerde, M., 1995. Statistical analysis of day-to-day variations in real-time traffic flow data. Transp. Res. Rec. 26–34.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning Representations by Back-propagating Errors. MIT Press, Cambridge, MA, USA.
Salehinejad, H., Talebi, S., 2008. A new ant algorithm based vehicle navigation system: A wireless networking approach. In: International Symposium on Telecommunications, IST 2008. IEEE., NJ, USA. pp. 36–41.
Salehinejad, H., Talebi, S., 2010. Dynamic fuzzy logic-ant colony system-based route selection system. Appl. Comput. Intell. Soft Comput.
Schrank, D., Eisele, B., Lomax, T., 2012. TTIs 2012 Urban Mobility Report. The Texas A&M Transportation Institute.
Shen, L., 2008. Freeway Travel Time Estimation and Prediction Using Dynamic Neural Networks. FIU Electronic Theses and Dissertations.
Siang Tan, K., Mat Isa, N.A., 2011. Color image segmentation using histogram thresholding–fuzzy c-means hybrid approach. Pattern Recognit. 44, 1–15.
Spalding, S., 2008. RACQ Congestion Roads Report: The Effects on Fuel Consumption and Vehicle Emissions. Technical Report. RACQ Vehicle Technologies Department.
Stützle, T., Hoos, H.H., 2000. Max–min ant system. Future Gener. Comput. Syst. 16, 889–914.
Suson, A., 2010. Dynamic Routing Using Ant-based Control (Master's thesis). Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology.
Tatomir, B., Rothkrantz, L., 2004. Dynamic traffic routing using ant based control. In: 2004 IEEE International Conference on Systems, Man and Cybernetics. IEEE. NJ, USA. pp. 3970–3975.
Tatomir, B., Rothkrantz, L., 2006. Hierarchical routing in traffic using swarm-intelligence. In: IEEE Intelligent Transportation Systems Conference, ITSC'06. IEEE. NJ, USA. pp. 230–235.
Tatomir, B., Rothkrantz, L.J., Suson, A.C., 2009. Travel time prediction for dynamic routing using ant based control. In: Proceedings of the 2009 Winter Simulation Conference (WSC). IEEE. NJ, USA. pp. 1069–1078.
Terr, D., 2004. Weighted Mean. From MathWorld-A Wolfram Web Resource, created by Eric W. Weisstein. 58, 13918–13927. 〈http://mathworld.wolfram.com/WeightedMean.html〉.
Theraulaz, G., Bonabeau, E., 1999. A brief history of stigmergy. Artif. Life 5, 97–116.
Tian, J., Ma, L., Yu, W., 2011. Ant colony optimization for wavelet-based image interpolation using a three-component exponential mixture model. Expert Syst. Appl. 38, 12514–12520.
Van Lint, J., Hoogendoorn, S., van Zuylen, H.J., 2005. Accurate freeway travel time prediction with state-space neural networks under missing data. Transp. Res. Part C: Emerg. Technol. 13, 347–369.
Wang, C., 2010. The Relationship Between Traffic Congestion and Road Accidents: an Econometric Approach Using GIS (Ph.D. thesis). © Chao Wang.
Wang, S., Djahel, S., McManis, J., 2013. A Hybrid Vehicular Re-routing Strategy with Dynamic Time Constraints for Road Traffic Congestion Avoidance.
Wedel, J.W., Schunemann, B., Radusch, I., 2009. V2x-based traffic congestion recognition and avoidance. In: 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks (ISPAN). IEEE. NJ, USA. pp. 637–641.
Yang, X.S., Cui, Z., Xiao, R., Gandomi, A.H., Karamanoglu, M., 2013. Swarm Intelligence and Bio-inspired Computation: Theory and Applications. Newnes, Amsterdam, Netherlands.
Yousefi, P., Zamani, R., 2013. The optimal routing of cars in the car navigation system by taking the combination of divide and conquer method and ant colony algorithm into consideration. Int. J. Mach. Learn. Comput. 3, 44–48.
Yue, H., Shao, C., 2007. Study on the application of A^* shortest path search algorithm in dynamic urban traffic. In: Third International Conference on Natural Computation, ICNC 2007. IEEE. NJ, USA. pp. 463–469.